

# 확장된 계층적 랜덤 그래프를 이용한 필기 한글 인식

김호연, 김진형  
한국과학기술원 전산학과

Handwritten Hangul Recognition using Extended Hierarchical Random Graph

HoYon Kim, JinHyung Kim  
Dept. of Computer Science, KAIST

## 요 약

본 논문에서는 계층적 랜덤 그래프를 이용한 필기 한글 인식 방법론을 제안한다. 한글은 다른 문자와 달리 기본 자소의 조합으로 이루어진 문자로서 2차원 평면상에 표현된다. 이러한 한글의 특성과 필기된 한글에서 나타나는 다양한 변형을 통계적으로 모델링하기 위해서 계층 그래프를 이용하였다. 특히, 계층 그래프의 최 하위 계층에서는 필기된 획의 변형을 흡수할 수 있도록 확장된 랜덤 그래프를 적용하였다. 제안된 모델은 통계적 모델이기 때문에 필기 데이터베이스로부터 모델의 파라미터를 구할 수 있다는 장점이 있다. 실험에서 제안된 모델을 필기 한글 인식 문제에 적용하여 자소간 접촉된 문자나 어느 정도의 흘려 쓴 문자도 잘 인식할 수 있음을 보였다.

## 1. 서론

필기 문자는 인쇄 문자와 달리 획의 접촉이나 흘림에 따라 문자 형태의 변형이 크기 때문에 인식이 어렵다. 특히 한글은 문자의 종류가 많기 때문에 인식의 복잡도가 숫자나 영어 낱자에 비해 매우 크며, 기본 자소들의 조합으로 글자가 이루어지기 때문에 글자간의 유사도가 한자에 비해 매우 높다. 필기 한글 인식기의 성능을 높이기 위해서는 이러한 한글의 특성을 적절히 반영할 필요가 있으며, 기존의 연구도 대부분 이러한 맥락에서 진행되어 왔다[5][6][7][8][9][10][13].

이처럼 한글의 특성을 반영하여 자소의 조합으로 인식하기 위해서는 먼저 자소 분할이라는 문제를 해결해야 한다. 그러나, 필기된 문자는 자소간의 접촉이나 흘림이 심하기 때문에 문자에 대한 정보(인식 결과) 없이 자소를 분할하기가 어렵다. 이러한 문제를 해결하기 위해서는 자소 기반 모델링을 하되 문자 인식과 자소 분할이 상호 보완적으로 수행되어야 한다.



그림 1.1: 흘려 쓴 필기 한글의 예

필기 문자는 그 특성상 필기도구로 평면상에 그려진 획과 이러한 획들이 2차원 평면상에 놓여진 공간적 위치 관계에 의해서 표현될 수 있다. 연속적으로 이어진 점들이 하나의 획을 구성하게 되며, 새로운 획을 쓰기 위해서 필기자는 필기도구를 들고 이동하여 새로운 위치에서 다음 획을 그리기 시작한다. 이와 같은 필기 문자의 생성 과정을 고려해 보면, 필기 문자를 모델링하기 위해서 적어도 2단계의 과정이 필요함을 알 수 있다. 더구나 한글은 기본 자소의 조합으로 이루어졌으므로, 필기 한글 문자는 계층적 모델을 이용하여 효과적으로 표현할 수 있다.

본 논문에서 제안하는 계층적 문자 모델은 획 자체를 모델링하는 단계와 각각의 획들의 공간적 위치 관계를

를 모델링하는 단계를 계층적으로 구분한 것이다. 즉, 계층 모델의 아래 단계에서는 필기 도구로 그려진 획 자체를 모델링하며, 윗 단계에서는 평면상에 위치하고 있는 각각의 획들과 자소들의 공간적 위치 관계를 단계별로 모델링 한다.

이와 관련된 기존의 연구로 계층적 속성 그래프를 한자 인식에 적용한 것[3]과 2단계 랜덤 그래프를 한자 인식에 적용한 연구[2]가 있다. 두 방법 모두 입력 문자를 계층적 그래프로 표현한 다음 이를 문자 모델과 매칭하였다. 그러나 필기 한글을 이용하여 이와 같은 계층 그래프를 만들 경우 많은 오류가 발생할 수 있다. 초성과 중성, 종성을 인식하지 않은 상태에서 이를 분할하여 계층 그래프를 만드는 과정에서 잘못된 분할이 일어날 가능성이 많기 때문이다. 또한 두 방법 모두 획을 기반으로 인식을 시도하였으나 획을 수평, 수직, 사선, 곡선 등으로 단순화하고, 획간의 관계도 몇 가지 형태로 단순화 시켰기 때문에 한글과 같이 유사한 문자에 적용할 경우 변별력이 떨어지게 된다.

본 논문에서는 이러한 단점을 보완하여 변별력을 높이고 한글의 특성을 반영한 계층적 랜덤 그래프를 이용하였다. 계층 그래프의 아래 단계는 다양한 형태의 연결된 획을 모델링하기 위해서 랜덤 그래프를 확장하여 이용하였다. 윗 단계는 획들의 공간적 인접성을 모델링하기 위해서는 랜덤 그래프를 계층적으로 적용하였다.

입력 문자로부터 만들어진 속성 그래프와 계층 그래프와의 매칭은 아래 단계의 획 모델부터 자소, 문자 모델의 순서로 이루어지며, 이 과정에서 가능한 모든 조합을 확률적으로 고려하여 매칭하기 때문에 최적의 매칭 결과를 얻을 수 있다.

제안된 한글 인식 시스템의 특징을 요약하면 다음과 같다.

- 획기반 인식
- 자소기반 인식
- 계층적 모델링
- 통계적 방법

## 2. 계층적 랜덤 그래프

계층적 랜덤 그래프[2]를 설명하기 앞서 랜덤 그래프의 정의를 살펴보면 다음과 같다.

### 2.1 랜덤 그래프

랜덤 그래프는 완전한 확률공간상에서 정의되며, 그 출력값이 그래프인 확률 모델로서 다음과 같이 정의할 수 있다[1].

정의 2.1: 랜덤 그래프  $R(R_N \cup R_E)$ 은 다음과 같은 4개의 튜플로 표현된다.

- $$R = (N, E, \mu, \delta)$$
- 1)  $R_N = \{ \alpha_1, \alpha_2, \dots, \alpha_n \}$ , 단  $\alpha_i$ 는 랜덤 노드로 랜덤 변수이다.
  - 2)  $R_E = \{ \beta_1, \beta_2, \dots, \beta_m \}$ , 단  $\beta_i$ 는 랜덤 아크로 랜덤 변수이다.
  - 3)  $N$ : 유한개의 노드들의 집합
  - 4)  $E: N \times N$ 의 부분집합으로  $N$ 에 속한 원소들의 순서쌍의 집합
  - 5)  $\gamma = (\mu, \delta)$ : 함수, 단  $\mu: N \rightarrow R_N, \delta: E \rightarrow R_E$ 는 각각에 대한 함수

그래프  $G = (N, E)$ 를 랜덤 그래프  $R$ 에서 출력 가능한 임의의 그래프라 하면,  $R$ 에서 그래프  $G$ 를 출력할 확률을 계산할 수 있다.  $\mu$ 와  $\delta$ 를 각각  $\mu$ 와  $\delta$ 의 역함수라 정의하면 주어진 매칭(monomorphism)  $\gamma$ 에 대해 랜덤 그래프  $R$ 에서  $G$ 의 출력 확률은 다음과 같은 식으로 구할 수 있다.

$$P_R(G, \gamma) = \prod_{\forall \alpha_i} \text{Prob}\{n = \mu'(\alpha_i), \alpha_i \text{ in } R_N\} \\ \times \prod_{\forall \alpha_i \forall \beta_j} \text{Prob}\{(a = \delta'(\beta_j), \beta_j \text{ in } R_E) \\ (n = \mu'(\alpha_i), \alpha_i \text{ in } R_N)\} \quad (\text{식 2.1})$$

단,  $n$ 은  $N$ 의 원소,  $a$ 는  $E$ 의 원소

식 2.1로부터 알 수 있는 것은 출력 확률(매칭확률)을 구하기 위해서는 먼저  $G$ 의 각 노드와 아크를 랜덤 그래프에 매칭하는 함수를 구하고, 이때 각 노드와 아크의 매칭 확률을 곱하면 된다는 것이다. 이러한 계산식은 랜

덤 그래프의 각 노드와 아크가 서로 독립적이라는 가정을 통하여 얻을 수 있다.

## 2.2 확장된 랜덤 그래프

기존의 랜덤 그래프는 아크와 노드들의 매칭이 일대일(one-to-one)로 되기 때문에 기본적으로 다대일(multiple-to-one) 매칭이 불가능하다. 따라서 확 매칭에 적용될 경우 매칭이 매우 제한적일 수밖에 없다. 이러한 단점을 개선하고 확 모델링에 적용하기 위해서 랜덤 그래프를 확장하였다.

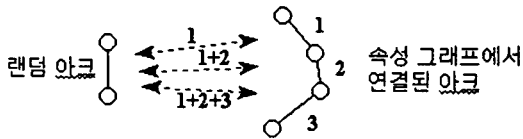


그림 2.1: 랜덤 아크의 다대일 매칭

확장된 랜덤 그래프에서는 일련의 아크가 하나의 아크에 매칭될 수 있다. 다시 말하면, 하나의 랜덤 아크의 출력으로 연결된 여러 개의 아크가 나올 수 있다는 것이다. 이를 위해서 랜덤 아크에는 매칭될 아크의 길이를 나타내는 랜덤 변수와 아크의 형태를 나타내는 랜덤 변수가 속성으로 추가된다. 확장된 랜덤 그래프의 매칭 알고리즘을 간단히 요약하면 다음과 같다.

### MatchGraph(A, N, a, n) :

;; 랜덤 그래프와 속성 그래프의 매칭 알고리즘

입력 :

랜덤 그래프의 현재 아크 A와 노드 N

속성 그래프의 현재 아크 a와 노드 n

알고리즘 :

1) N에 연결된 아크(A)가 1개뿐이면 매칭결과 저장 후 return;

2) A나 a가 이미 방문 되었으면 매칭 실패, return;

3) 노드 N과 노드 n을 중심으로 A와 a로부터 시계방향으로 각각의 부분 그래프를 매칭

N에 연결된 아크 A'과 n에 연결된 아크 a'의

매칭확률 계산

MatchGraph( ) ;; A', a'과 연결된 나머지 부분 그래프를 매칭

MatchGraph2( ) ;; multiple-to-one 매칭을 위해서 아크 A는 그대로 두고 속성 그래프의 아크 a와 노드 n만을 바꾸어 부분 그래프 매칭

(나머지는 MatchGraph( )와 동일)

4) 매칭된 각 부분 그래프(노드와 아크들)의 매칭 결과를 통합

5) 현재까지의 부분 그래프 매칭 확률을 return;

위의 알고리즘은 랜덤 그래프를 속성 그래프와 매칭하기 위한 알고리즘으로서 종료 조건에 따라 속성 그래프의 전체 혹은 부분그래프와 매칭이 가능하다. 이 알고리즘은 매칭할 때 매칭이 시작되는 점을 중심으로 recursive하게 매칭을 수행한다. 가능한 조합을 다 계산하기 위해서 매칭시 랜덤 그래프의 시작노드는 고정시키고 속성 그래프의 시작노드를 바꾸어 가면서 매칭한다. 각 아크의 매칭확률은 해당 랜덤 아크에서 입력 아크 속성에 대한 확률 분포를 이용하여 구할 수 있으며, 노드의 매칭확률도 마찬가지로 구할 수 있다. 이 알고리즘을 이용하면 매칭이 가장 잘 될 때의 매칭과 그때의 매칭 확률을 얻을 수 있다.

## 2.3 계층적 랜덤 그래프

계층적 랜덤 그래프[2]란 랜덤 그래프의 랜덤 노드의 출력이 다시 랜덤 그래프로 정의된 그래프를 말한다. 본 논문에서 사용된 계층적 랜덤 그래프는 3계층으로 구성된다. 계층 그래프의 첫 계층(최하위 계층)은 확 모델링을 위해 확장된 랜덤 그래프로서 확의 방향, 길이 등을 랜덤 아크와 랜덤 노드의 속성으로 갖는다. 두번째 계층은 자소를 나타내기 위한 것으로 랜덤 노드의 속성으로 갖는다. 마지막 계층은 한글 문자를 나타내는 계층으로서 한글 자모의 결합에 따라 6가지 형태의 랜덤 그래프가 존재한다.

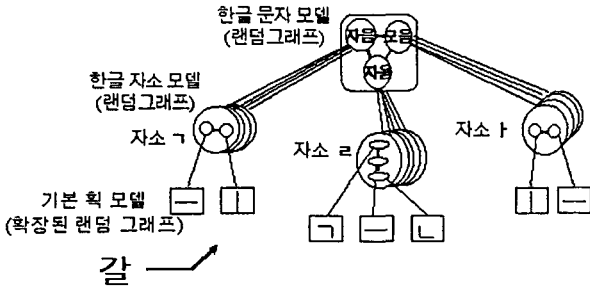


그림 2.2: 계층적 한글 모델의 예

### 3. 계층적 한글 모델

필기된 한글은 획들의 조합으로 자소를 이루고, 자소들의 조합으로 문자를 이루는 계층적 구조로 되어 있다. 본 절에서는 이러한 관점에서 각 계층에서의 모델링 방법을 설명한다.

#### 3.1 획 모델

획 모델은 필기된 기본 획을 나타내기 위한 모델로서, 확장된 랜덤 그래프를 이용하여 연결되어 있는 획들을 모델링한 것이다. 제안된 인식기에서는 ‘|’, ‘-’, ‘/’, ‘\’, ‘ㄱ’, ‘ㄴ’, ‘ㄷ’, ‘ㄱ’, ‘ㄴ’, ‘ㄷ’, ‘ㅇ’, 등 한글에 자주 나타나는 기본적인 획 모양을 모델링하기 위한 획 모델이 있다.

획 모델의 예를 들면 ‘ㄱ’ 모델의 경우 그림 3.1 과 같이 두 개의 랜덤 아크와 세 개의 랜덤 노드로 표현된다. 이때, 랜덤 아크는 해당 부분에 매칭될 획의 방향 코드와 길이 코드에 대한 확률 분포를 갖으며, 랜덤 노드는 연결된 아크들의 개수와 연결각에 대한 확률 분포를 갖는다. 확장된 랜덤 그래프는 하나의 랜덤 아크가 여러 개의 아크와 매칭이 가능하기 때문에 그림 3.2 에서와 같이 길이와 방향이 다른 여러 형태의 획이 하나의 랜덤 그래프 모델로 표현이 가능하다.

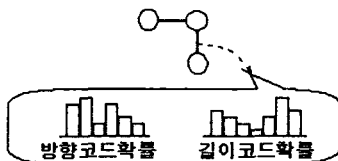


그림 3.1: ‘ㄱ’ 모델에서 랜덤 아크의 확률분포 예



그림 3.2: ‘ㄱ’ 모델에 매칭 가능한 속성 그래프

#### 3.2 자소 모델

자소 모델은 노드의 출력이 획 모델이고 아크의 출력은 획 모델간의 위치관계를 나타내는 랜덤 그래프이다. 예를 들어 자소 모델 ‘ㄴ’의 경우 그림 3.3 에서와 같이 각 노드는 기본획 ‘ㄱ’, ‘-’, ‘ㄴ’을 나타내며 아크는 연결된 기본획들의 연결상태를 확률 분포로 표현하고 있다. 두 노드의 연결상태를 모델링하기 위해서 각각의 중점을 연결하는 방향벡터를 고려하기도 하고, 자소의 특성을 효과적으로 반영하기 위해서 미리 정해진 각 노드의 특징점들을 연결하는 방향벡터를 확률분포로 나타내어 이용하기도 한다.

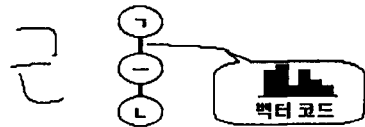


그림 3.3: 자소모델 ‘ㄴ’에서 벡터코드 확률 분포의 예

#### 3.3 문자 모델

문자 모델은 자소 모델과 마찬가지로 랜덤 그래프로 되어있으나 노드의 출력이 자소라는 차이가 있다. 마찬가지로 문자 모델의 아크는 해당 자소간의 연결상태를 출력값으로 갖는다. 한글은 자모의 위치에 따라 조합형태를 크게 6 가지로 나눌 수 있으므로 6 개의 문자 모델로 모든 한글에 대한 표현이 가능하다.

FC VV	FC HV	FC VV	FC VV	FC HV	FC HV VV	FC: 초성 (first consonant)
Type1	Type2	Type3	Type4	Type5	Type6	VV: 중 모음 (vertical vowel)
			LC	LC	LC	LC: 종성 (last consonant)
						HV: 횡 모음 (horizontal vowel)

그림 3.4: 한글의 6 가지 조합형태

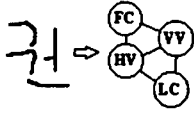


그림 3.5: Type6 한글 문자 '권'과 매칭되는 랜덤 그래프의 예

## 4. 인식 시스템

통계적 접근 방법을 이용할 경우 인식 문제는 주어진 입력에 대해 가장 매칭 확률이 높은 모델을 찾는 문제라 할 수 있다. Bayes 정리를 이용하여 이를 수식으로 나타내면 다음과 같다.

$$\begin{aligned} & \operatorname{argmax}_{(M, G)} P_M(G^*|X) \\ & = \operatorname{argmax}_{(M, G)} P_M(X|G^*)P(G^*) \end{aligned}$$

단,  $X$ : 입력

$G^*$ : 계층 그래프

$M$ : 입력 아크와 노드로부터 모델로의 함수

위 식에서  $P(G^*)$ 는 미리 알고 있거나 동일하다고 가정하면,  $P_M(X|G^*)$ 만 구하면 되는데 이 값은 모델  $G^*$ 에 입력  $X$ 가 매칭될(나타날) 확률이다. 랜덤 그래프에서 이 값은 그래프의 모든 아크와 노드의 매칭 확률을 곱하여 얻을 수 있다. 따라서, 입력 문자 영상으로부터 인식 결과를 얻기 위해서는 먼저 입력 문자 영상을 속성 그래프로 변환하여 표현한 뒤, 이를 계층 그래프와 매칭하여 최적 매칭된 매칭 함수를 구하면 된다. 이와 같이 인식 단계는 속성 그래프를 만드는 전처리 단계와 이를 모델과 매칭하는 인식 단계로 나눌 수 있다.

### 4.1 전처리 단계

전처리 단계의 목적은 문자 영상으로부터 정확한 획을 추출하고, 추출된 획을 이용하여 인식 모델과 매칭이 가능한 속성 그래프를 만드는 것이다.

먼저 획 추출 과정을 간단히 설명하면 다음과 같다. 잡영을 제거하기 위해서  $3 \times 3$  Gaussian 마스크를 이용하여 문자 영상을 평활화한 뒤, 히스토그램의 엔트로피를 이용한 임계치 결정기법[19]을 적용하여 문자 영상을 이진화한다. 이러한 방법으로 미세한 잡영이 제거된 문자 영상에 세션화 알고리즘을 적용하면 문자의 골격선을 얻을 수 있다. 다음으로 끝점이나 분기점을 기준으로 연

결된 점을 따라가면 획을 추출할 수 있다.

추출된 획을 이용하여 속성 그래프를 만들기 위해서 우선 각 획은 그림과 같이 양 끝점에서 가장 먼 점이 미리 정해진 임계값보다 작을 때까지 반복해서 분할하는 방식으로 직선으로 근사된다. 각각의 직선은 속성 그래프에서 하나의 아크로 표현이 되며 직선의 끝점은 노드로 표현된다.

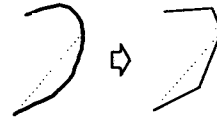


그림 4.1: 획의 직선 근사 과정

문자를 표현하기 위해서 적용된 속성 그래프[3] (Attributed Graph)  $G = \langle N, E \rangle$ 는 다음과 같이 정의할 수 있다.

정의 3.1 :  $n$ 개의 노드와 이를 연결하는  $m$ 개의 아크를 갖는 속성 그래프  $G$ 는 노드들의 집합  $N$ 과 아크의 집합  $E$ 로 표현된다.

$$G = \langle N, E \rangle$$

$$N = \{ v_i \mid v_i \in G \text{ 속한 노드}, 1 \leq i \leq n \}$$

$$E = \{ a_{i(j,k)} \mid a_{i(j,k)} \text{ 는 } v_j \text{ 와 } v_k \text{ 를 연결하는 아크}, 1 \leq i \leq m \}$$

이때, 각 아크와 노드는 속성(attribute)을 갖게 되는데, 각 아크의 속성으로는 해당 직선성분의 길이와 기울기가 있고, 노드의 속성으로는 아크들의 연결각, 아크의 수, 노드의 특성 등이 있다. 노드의 특성이란 연결된 아크들의 각도에 따라서 노드를 분할점, 비분할점으로 정의한 것으로서, 매칭의 효율을 높이기 위해서 그림의 c)에서와 같이 끝점이나 분기점, 꺾인 각도가 큰 점을 분할 점으로 정하여 매칭시 이러한 분할점에서만 매칭이 끝날 수 있도록 제한하였다.

아크와 노드의 속성은 이산확률분포를 갖는 모델과의 매칭을 위해서 미리 양자화된 값으로 표현된다. 다시 말해서 직선의 기울기와 꺾인 각 등은 16 방향 코드값으로 바뀌게 되고, 직선의 길이나 떨어진 거리는 글자 높이의  $1/12$ 를 단위로 한 코드값으로 바뀌어 표현된다. 여

기서 주의할 점은 방향코드가 나타내는 방향은 획이 쓰여진 방향을 의미하는 것이 아니라 매칭시에 매칭의 기준이 되는 점으로부터의 방향을 의미한다. 그림 4.2는 매칭의 시작점에 따라 달라지는 방향코드를 나타낸다.

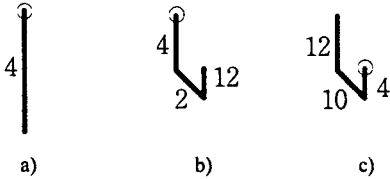


그림 4.2: 매칭 기준점에 따른 방향 코드의 변화

전처리 과정에서 획 추출시 방향이 같은 획을 미리 연결할 때 생기는 오류를 없애기 위해서 그림 c)처럼 전처리 단계에서 1, 2 혹은 4, 6 획을 미리 연결하지 않고 인식 도중 최적 매칭을 찾을 수 있도록 하였다.

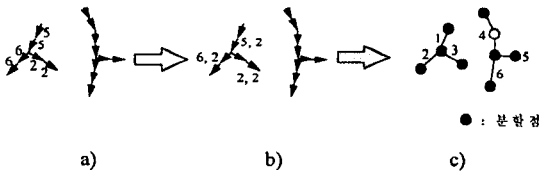


그림 4.3: 기울기 코드로 변환된 예 : a) 기울기 코드, b) 기울기 코드와 해당 획의 길이 코드, c) 속성 그래프

## 4.2 인식 단계

계층 그래프의 매칭은 획 매칭, 자소 매칭, 문자 매칭 순서로 아래 계층부터 이루어진다.

### 4.2.1 획 매칭

문자 영상에 전처리 과정을 적용하여 만들어진 속성 그래프를 입력 그래프라고 명하기로 하자. 획 매칭이란 앞 절에서 정의한 획 모델들과 입력 그래프를 매칭하여 자소를 이룰 가능성이 있는 모든 기본 획 후보를 추출하는 과정을 말한다. 획 매칭 알고리즘은 확장된 랜덤 그래프의 매칭 알고리즘과 동일하며, 결과적으로 입력 그래프의 부분 그래프들 중에서 획 모델과 높은 확률로 매칭되는 그래프들이 매칭 확률과 함께 매칭 결과로 얻

어진다. 이와 같이 매칭된 그래프는 다음 단계에서 자소를 얻기 위한 자소 매칭에 사용된다.

획 매칭시에 매칭의 복잡도를 줄이기 위해서 미리 지정된 분할점에서만 부분 그래프로의 분할이 가능하도록 하였으며, 또한 모든 아크가 2 차원 평면상에 놓여있다는 특성을 이용하여 획 매칭의 조합을 줄일 수 있었다.

### Bottom-up 매칭

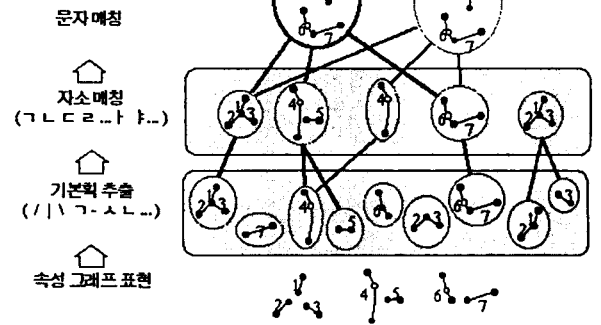


그림 4.4: 계층 그래프 매칭의 중간 과정

### 4.2.2 자소 매칭

자소 매칭 단계는 획 매칭에서 생성된 기본 획 후보들을 조합하여 자소를 생성하는 단계로서 앞 단계에서 생성된 획 후보들을 자소 모델에 조합하여 매칭 확률이 높은 자소 후보를 찾는 단계이다. 자소 매칭 단계에서는 매칭 효율을 높이기 위해서 매칭될 후보 자소의 개수를 제한하여 매칭 확률이 낮은 것과, 획 후보들 간의 이웃 정보를 이용하여 하나의 자소를 이룰 수 없는 획들(이웃하지 않은 획들)의 매칭은 미리 제거하였다.

### 4.2.3 문자 매칭

앞 단계에서 매칭된 자소 후보들을 조합하여 문자를 생성하는 단계로서 입력 그래프의 모든 아크가 중복되지 않고 매칭에 포함될 수 있도록 매칭한다. 또한 매칭되지 않은 부분은 남은 길이에 비례하게 벌점(penalty)을 부여함으로써 가능한 많은 획이 매칭될 수 있도록 하되, 잡영에 해당하는 획은 생략하여 약간 낮은 확률로 매칭

이 가능하도록 하였다. 이처럼 가능한 자소 후보들을 조합하여 매칭을 수행하기 때문에 자소 분할을 미리 함으로써 발생하는 분할 오류를 피할 수 있었다.

## 5. 실험결과

제한한 필기 한글 인식 시스템의 성능 평가를 위하여 PE 92 데이터를 이용하였다. 계층 그래프 모델의 구조는 한글에 대한 경험적 지식을 이용하여 설계하였으며, 모델의 파라미터도 시스템 설계자가 모델 에디터를 이용하여 지정하였다. 초기 모델을 디자인할 때 모델 간의 변별력을 높이기 위해서 각 자소 혹은 획 모델에서 중요한 특징점을 선택하고 이들의 위치관계를 확률 분포로 모델링 하였다.

모델의 개수는 ‘ㄱ’과 같이 모양이 거의 비슷한 자소의 경우 1개, ‘ㄹ’이나 ‘ㄴ’, ‘ㄷ’과 같이 필기자에 따라 변형이 큰 자소의 경우 2~3 개로 자소당 1~3 개로 정하였다.

실험 환경은 Sparc 20 에서 C++로 구현하였으며 인식 시간은 획의 개수에 따라 차이가 있으나 평균적으로 초당 0.5~2 자의 인식 성능을 보였다. 비교적 인식 시간이 오래 걸리지만 이는 매칭 알고리즘을 좀더 효율적으로 구현할 경우 줄일 수 있을 것으로 생각된다. 인식 실험에서 PE92 데이터의 앞부분 10 개 파일(23500 자)에 대해서 인식율 71% (2 순위까지 : 80.7%, 5 순위까지 : 85.2%)를 얻었다. 실험에 사용된 모델은 수작업으로 만들었다.

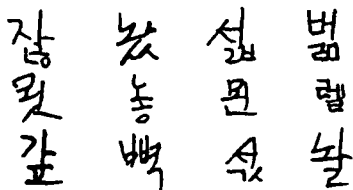


그림 5.1: 필기 한글 데이터의 예 (인식된 글자)

그림 5.1, 5.2 는 인식에 성공한 문자의 예이다. 제안된 시스템이 자소 기반 인식 시스템이기 때문에 인식이 끝나면 그림 5.2 와 같이 자소의 분할 정보를 동시에 얻을 수 있고, 그림 5.3 과 같이 빠침이나 잡영에 의한 획을

무시하고 인식이 가능하다는 장점이 있다.

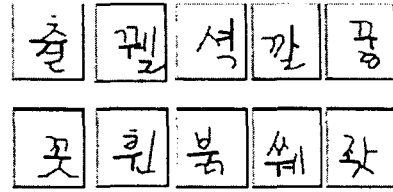


그림 5.2: 인식되어 자소 분할에 성공한 문자의 예

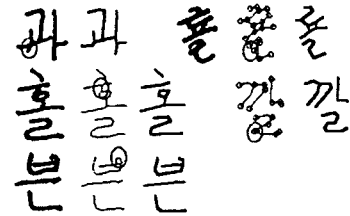


그림 5.3: 잡영에 무관하게 인식에 성공한 예

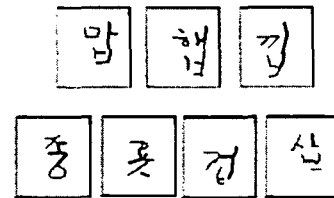


그림 5.4: 잘못된 분할 및 오인식된 예

## 6. 요약 및 결론

본 논문에서는 필기 한글 인식을 위한 자소 기반 인식기를 개발하였다. 자소 기반 인식기는 한글의 체자원리를 반영한 것으로 적은 수의 자소 모델로 모든 한글 문자를 인식할 수 있다는 장점이 있다. 이를 위해서 통계적 모델인 계층적 랜덤 그래프 모델을 도입하여 한글을 모델링 하였다.

계층적 랜덤 그래프는 아래 단계에서 필기된 문자의 획을 모델링하고 위 단계에서 이러한 획간의 연결관계와 위치관계를 모델링하도록 구성되었으며, 필기 형태에 대한 특별한 제약이 없으므로 영어나 숫자, 한글 등 다양한 필기 문자 모델로도 응용될 수 있다. 특히 획 모델링을 위해 확장된 랜덤 그래프는 필기된 획의 변형을 흡수할 수 있으므로 적은 수의 모델로 다양한 획의 표

현이 가능하다는 장점이 있다.

제안된 모델에서 문자의 구조적 정보는 그래프의 구조에 표현되며 자소나 획의 다양한 변형은 각 랜덤 노드와 랜덤 아크에 확률 분포로 나타내어진다. 이러한 확률 분포는 학습 데이터로부터 통계적으로 구하는 것이 바람직하다. 그러나 이를 위해서는 자소 단위로 분할된 다량의 데이터가 필요하기 때문에 인식 실험에는 시스템 설계자가 수작업으로 만든 모델을 사용하였다. 만일 모델 파라미터가 실제 데이터를 더 잘 반영하도록 학습에 의해 조정된다면 인식 성능은 좀더 향상될 것이다. 현재 이를 위한 자동 학습 알고리즘에 대한 연구가 진행 중이다.

인식기 성능 평가를 위해 PE92 데이터베이스를 이용한 인식 실험에서, 제안된 방법이 흘려 쓴 문자나 자소 간 접촉이 있는 문자도 인식할 수 있음을 알 수 있었다. 제안된 인식기의 성능을 개선하기 위해서는 혼동되는 문자를 구분하기 위한 특징을 추가하거나 자소 쌍에서 나타나는 특징들을 이용한 후처리 등의 방법을 도입해서 모델의 변별력을 높여야 할 것이다. 또한 획 추출시의 오류를 줄이기 위해서는 전처리 단계를 좀더 보완해야 한다.

## 참고문헌

[1] A. K. C. Wong and D. E. Ghahraman. Random graphs: structural-contextual dichotomy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-2(4):341--348, Jul. 1980.

[2] L. H. Chen and J. R. Lieh, "Handwritten character recognition using a 2-layer random graph model by relaxation matching," *Pattern Recognition*, Vol.2, No.11, pp. 1189-1205, 1990.

[3] S. W. Lu, Y. Ren and C. Y. Suen, "Hierarchical attributed graph representation and recognition of handwritten chinese characters," *Pattern Recognition*, Vol.24, No.7, pp.617-632, 1991.

[4] J. Rocha and T. Pavlidis, "A shape analysis model with applications to a character recognition system," *IEEE Tr. on Pattern Analysis and Machine Intelligence*, Vol.16, No.4, pp.393-404, April, 1994.

[5] 박종욱, 이주근, Shape 패턴에 의한 필기체 한글 인식.

대한전자공학회지, 제 22 권, 제 5 호, pp.1--9, 1985 년

[6] 이승호, 김진형, 한글의 구조적 인식을 위한 자획 추출에 관한 연구. 한국정보과학회 학술발표논문집, 제 14 권, 제 2 호, pp.151--154, 1987 년

[7] 김태균, T. Agui, M. Nakajima, Stroke 조합에 의한 필기체 한글의 표현과 인식. 대한전자공학회지, 제 25 권, 제 1 호, pp.18--26, 1988 년

[8] 이은주, 권오석, 김태균, 필기체 한글에서 자모 분리와 인식. 한국정보과학회논문지, 제 15 권, 제 6 호, pp.526--534, 1988 년, 12 월

[9] 강희석, 권오석, 오길록, 퍼지척도에 의한 필기체 한글의 자모분리. 한국정보과학회 학술발표 논문집, 제 18 권, 제 2 권, pp.297--300, 1991 년

[10] Y. B. Kwon K. Y. Lee and Y. B. Lee. A novel Hanguil recognition algorithm based on stroke extraction. In *Proc. ICDAR 91*, volume 1, pages 272--280, 1991.

[11] H. S. Park S. W. Lee and Y. Y. Tang. Translation-, scale-, and rotation-invariant recognition of Hanguil characters with transformation ring projection. In *Proc. ICDAR 91*, pages 829--836, 1991.

[12] 전윤호, 최종호, 가중치를 공유하는 신경회로망을 이용한 필기체 한글문자 인식. 대한전자공학회지, 제 28 권, 제 8 호, pp.683--691, 1991 년, 8 월

[13] 김형래등, CFG 방법을 이용한 필기체 한글에서의 자소 추출과 인식에 관한 연구. 대한전자공학회지, 제 29B 권, 제 9 호, pp.655--663, 1992 년, 9 월

[14] 박희선, 이성환, 은닉 마르코프 모델을 이용한 필기체 한글의 오프라인 인식. 한국정보과학회 학술발표 논문집 제 20 권, 제 6 호, pp.890--902, 1993 년, 6 월

[15] J. H. Kim. A unified network-based approach for on-line recognition of cursive handwritings in multiple languages. In *Proceedings of the 1994 International Conference on Computer Processing Oriental Languages*, pages 10--13, Taejon, May 1994.

[16] H. J. Kim and P. K. Kim. Recognition of off-line handwritten Korean characters. *Pattern Recognition*, 29(2):245--254, 1996.

[17] J. N. Kapur, A New Method for Gray-Level Picture Thresholding Using the Entropy of the Histogram, *Computer Vision, Graphics, and Image Processing* 29, 273-285, 1985